AFRL-IF-RS-TR-2003-218
Final Technical Report
September 2003

EFFECTIVE COORDINATION OF MULTIPLE INTELLIGENT AGENTS FOR COMMAND AND CONTROL

Carnegie Mellon University

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J379

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

AFRL-IF-RS-TR-2003-218 has been reviewed and is approved for publication.

APPROVED:

/s/

NANCY A. KOZIARZ
Project Engineer

FOR THE DIRECTOR:

/s/

JAMES W. CUSACK, Chief
Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	SEPTEMBER 2003	Final Apr 98 – Jan 03

4. TITLE AND SUBTITLE
EFFECTIVE COORDINATION OF MULTIPLE INTELLIGENT AGENTS FOR COMMAND AND CONTROL

5. FUNDING NUMBERS
C - F30602-98-2-0138
PE - 63760E
PR - AGEN
TA - T0
WU - 09

6. AUTHOR(S)
Katia Sycara, Joseph Giampapa, and Michael D. Rectenwald

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Carnegie Mellon University
The Robotics Institute
5000 Forbes Avenue
Pittsburgh Pennsylvania

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFSF
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2003-218

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Nancy A. Koziarz/IFSF/(315) 330-2828/ Nancy.Koziarz@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***
RETSINA (REusable Task-based System of Intelligent Network Agents) is an open Multi-Agent System (MAS) in which heterogeneous agents engage in relations with the support of distributed infrastructure services. The goal of RETSINA project has been to provide the necessary infrastructure and agent types to allow an open system of agents whose interactions are facilitated rather than managed by infrastructure components. Another goal has been to create autonomous software agents functioning robustly in distributed environments, agents that are reusable in different application contexts, and that respond intelligently to changes in their environments. This development of a flexible, extensible and decentralized MAS is consistent with the shared vision of an interactive World Wide Web of Services (WWWS), a dynamic web where software agents act autonomously and cooperatively to effect changes in their environments, performing numerous and complex tasks for their human counterparts. RETSINA is scalable system designed to support individual users and teams of users in utilizing the results of information gathering and fusion for decision support, task management, cooperative interaction, and many other functions, in an open and dynamic environment, eventually to encompass the World Wide Web. In RETSINA, multiple agents cooperate to process user requests appropriately and flexibly, matching them to information sources that may be distributed over the Internet and other accessible information environments, such as intranets or other databases.

14. SUBJECT TERMS
Multi-Agent Systems, Autonomous Agents

15. NUMBER OF PAGES
16

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

List of Figures

1 Introduction

RETSINA (REusable Task-based System of Intelligent Network Agents) is an open Multi-Agent System (MAS) in which heterogeneous agents engage in relations with the support of distributed infrastructure services. The goal of RETSINA project has been to provide the necessary infrastructure and agent types to allow an open system of agents whose interactions are facilitated rather than managed by infrastructure components. Agent relations in RETSINA are conceived of as non-hierarchical, decentralized and based on peer-to-peer communications. The goal of RETSINA has been to create autonomous software agents functioning robustly in distributed environments, agents that are reusable in different application contexts, and that respond intelligently to changes in their environments. This development of a flexible, extensible and decentralized MAS is consistent with the shared vision of an interactive World Wide Web of Services (WWWS), a dynamic web where software agents act autonomously and cooperatively to effect changes in their environments, performing numerous and complex tasks for their human counterparts.

RETSINA is scalable system designed to support individual users and teams of users in utilizing the results of information gathering and fusion for *decision support, task management, cooperative interaction,* and many other functions, in an open and dynamic environment, eventually to encompass the World Wide Web. In RETSINA, multiple agents cooperate to process user requests appropriately and flexibly, matching them to information sources that may be distributed over the Internet and other accessible information environments, such as intranets or other databases. RETSINA supports many different specialized agents. Agents represent users (interface agents), the tasks that fused information is meant to contribute to (task agents), and the information resources (information agents).

In RETSINA, we define the following general agent types (See Figure 1):

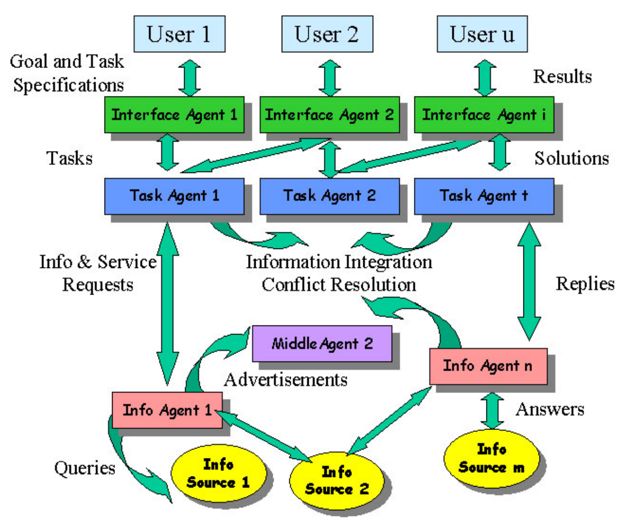

Figure 1: RETSINA MAS/ AGENT TYPES

Interface agents receive the user requests. These agents use knowledge of the domain and the user role and functionality to help the user formulate and customize his/her information requests. They also plan appropriate interactions with other agents on the user's behalf. Interface agents can also *learn to anticipate* user information needs and *proactively* pre-position information that the user has been observed in the past to need.

Task agents have knowledge of the task domain and also have planning abilities. Utilizing task models and planning, they are able to plan specific information-gathering goals or inferencing tasks. Task agents can decompose high level information requests to lower level tasks, and form plans for how to execute the information gathering subtasks, find (through middle agents) and query the appropriate information sources, and coordinate the query execution and composition of the query results. Task agents use planning mechanisms that combine planning and execution of the information gathering and inferencing tasks.

RETSINA agents utilize the Hierarchical Task Network (HTN) formalism. Planning is necessary, and it adaptively considers the current operational environment. For example, some information sources may be present at one time but absent at another time (e.g., a web source could be down or its link broken). Planning includes providing back-up information sources available for such contingencies.

Information agents: each information agent wraps an information source with a communications module (see Section 2) and knows the particular details of how to interact with the source to answer a query. In addition to one-time access to information, an information agent can be given event monitoring and notification triggers called monitoring queries. For example, "provide me the price of IBM stock every 5 minutes", or "provide me the price of IBM stock if it goes above a particular threshold."

Middle agents provide a level of indirection in information processing. They act as intelligent registries of agents and agent capabilities. They allow service/agent Discovery (see Part II) and lookup by semantically matching information needs of requester agents (or humans) with available information resources so that the requests can be routed to the available resources.

Given the agent types and infrastructure components of RETSINA, the vision of a web of services--where tasks are coordinated and provided by interoperable agents across many domains of data and knowledge bases--becomes a reachable goal.

1.1 Organization of the Report

This report provides a survey of some of the most substantial published research contributing to the RETSINA model of Multi-Agent Systems. It is intended as an introduction to Multi-Agent Systems, and particularly, the RETSINA system. The reader should refer to the bibliography for the detailed discussions of the particular topics.

The above introductory section on the RETSINA MAS Infrastructure provides the basic outline. The following sections correspond to the components of the RETSINA MAS Infrastructure as discussed in detail in [1].

2 The Communications Infrastructure

Peer-to-peer communications and multicast Discovery of agents and services contribute to the RETSINA Multi-Agent System (MAS) as a dynamic, extensible and robust architecture for the proliferation of agent systems throughout the World Wide Web [2].

The Communicator and Discovery go a long way toward the sought-after goal of a dynamic, interoperable community of agents who will perform tasks currently performed by human users, while combining to produce results that human users would not or could not produce.

Together, the Communicator and Discovery provide for several major types of agent interaction: direct, peer-to-peer, and multicast communications. These elements are generally recognized as essential components of the revolutionary web of agent services.

2.1 The Communicator

The Communicator supports connectivity of agents as both clients and servers, in peer-to-peer communications. Treating the underlying agent structure as a black box, RETSINA developed the Communicator as a plug-in, reusable module that can be wrapped around agents of any types, or around legacy software that thereby becomes "agentified." This lack of specificity as to the underlying agent modules allows communications between otherwise incommensurate, heterogeneous agent types, whose performance may depend upon different languages. RETSINA does not specify the Agent Communications Language (ACL) of the message content that agents exchange through the communications module, but only of the outer envelope, which is the communications module itself. Further, the Communicator can be easily adapted to FIPA or other ACL standards. Delimiting the function of the communications module to basic queries was the first step in making possible the kind of flexible, open architecture represented by RETSINA [3].

2.2 Discovery

In order to allow agents to locate other agents and distributed infrastructure, the RETSINA MAS has implement Discovery protocols that utilize multicast communications, a hybrid form of messaging combining broadcast and peer-to-peer methods. Given the multicast messaging of Discovery, agents and infrastructure services announce their presence and availability to the system and discover the existence and availability of other agents and infrastructure components [4].

Discovery has been implemented to the level of agent registration with the "white pages" directory of agents, known in RETSINA as Agent Name Servers (ANS). ANS Discovery works at both the server and client levels of agent and infrastructure communications. Agents, when signing on to the system, send out multicast queries for available ANSs. Thus, agents are asynchronously registered with multiple directories on the basis of their own "alive" announcements. Further, ANSs share ("push") their registration directories with each other, further disseminating the knowledge of agents throughout the distributed environments. Agent lookups of other agents are also shared by ANSs, giving "long distance" lookup capabilities to agent communities. Agents also keep registries of other agent names and locations, acting as caches of agent name and location information for frequently contacted peers.

Among the advantages derived from Discovery are system robustness and recoverability, agent survivability in the absence of known infrastructure, and the dramatically enhanced knowledge sharing of and amongst agents. With Discovery, agents are able to survive startup when an ANS is unavailable. Agents remain alive and receive announcements from ANSs as they become available. As ANSs share the information about their agent registrations with other ANSs, the possibilities of reaching other agents in widely distributed networks are greatly increased. As agents are registered with multiple ANSs, the chances of system failure to due to disappearance of one ANS are greatly reduced.

Further, Discovery has been extended to wide area networks (WAN) using the underlying lookup protocol of the Gnutella peer-to-peer file-sharing service. The basic Gnutella discovery schema is enhanced in RETSINA by the addition of parameters that allow agents to define and locate "islands of interests" where agents seeking similar services and interactions can gather for easier access.

While Discovery was initially implemented for name-location registrations and lookup (i.e., applied to ANS infrastructure), it has since been extended to work with other infrastructure components, including Matchmaking (see Section 3). Discovery is being adapted for other infrastructure and agent services. The goal is a completely dynamic and automatically discoverable MAS infrastructure.

3 Agent Communication Language (ACL), Middle Agents and RETSINA Infrastructure on the CoABS GRID

Ideally, the extended agent community will be comprised of a seamlessly interconnected web of requestors, providers, and middle agents who match requestors to providers. But agents who need to share information or services must first be able to find each other; they must have a common language for specifying their own and other agents' capabilities. Further, they must ascribe common meanings to the concepts that their languages specify, to the terms that they use. That is, for agents to communicate and cooperate, they must share an ontology.

RETSINA has addressed each of these related problems in agent architecture with infrastructure components that enable the matching of agents to each other based on agent capabilities expressed in a capabilities language called LARKS (**L**anguage for **A**dvertisement and **R**equest for **K**nowledge **S**haring), that store and reference these capabilities in Middle Agents called Matchmakers, and that describe the meaning of words/concepts of local ontologies, using LARKS [5].

In addition, the RETSINA Lab has contributed infrastructure components to the Globalinfotek CoABS GRID. A long-standing result of MAS research has been that there is no one coordination algorithm that is suitable for all tasks and environments. Hence, at the CMU/Sycara Lab, we have developed, demonstrated and evaluated a flexible agent coordination infrastructure that allows experimentation with diverse and realistic command and control scenarios. By using the RETSINA MAS infrastructure, it is possible to rapidly achieve a critical mass of agents and types of agents so that new and innovative scalability and coordination experiments can be performed.

Given RETSINA's dynamic and scalable Middle Agent infrastructure, including our ANS, Matchmaker, InterOperator (see Section 4) and the use of Discovery, our MAS is able to work collaboratively with other agent projects and simulations, including ALP, MIATA, CoAX and Teamcore.

3.1 Middle Agents/ACL Infrastructure

An underlying premise of the agent community is the notion that each agent provides a service or services to other members of the community. For example, a weather agent provides weather services, a flight agent provides flight information, a hotel agent provides hotel information and a reservation may recombine the information provided by these several agents for delivery to an interface agent. In order to cooperate and communicate in such a scenario, each agent must describe declaratively the service it provides in a high level semantic description, known as its capability advertisement. The capability advertisement is written in LARKS and communicated to a Matchmaker agent. Matchmakers store this information and match stored advertisements with requests for services. The information returned also includes the agent name and location, so that agents can begin to communicate on a peer-to-peer basis after they become known to one another.

RETSINA has identified a number of Middle Agent types, and has developed its Matchmaker agent type to provide a method of middle agent interaction that we believe to be optimal for the kind of open, decentralized, dynamic, flexible and robust MAS we envision [6, 7, 8, 9].[1] Our experimental evaluation of different types of middle agents show that different middle agent types promote different performance tradeoffs and can be suitable for different applications.

4 Security Infrastructure

One of the important issues facing the developers of open Multi-Agent Systems such as RETSINA is the issue of security. As different agent types assume important transaction roles for their human counterparts--such as stock portfolio management, auctions or other interactions that depend on trusted participants (see Section 6: Applications)--the need for security in an MAS becomes quite apparent. The security threats that agents, users and infrastructure might face are numerous. For example, agents of malevolent users may try to unregister their competitors from ANSs and Matchmakers. Agents may eavesdrop on supposedly private communications, or spoof other agents and the human deployers of agents. Agents may register with an ANS while claiming to be at given host and port, when in fact existing on different host and port. Agents may take part in a transaction; yet they might fail to meet their own obligations (such as making payments for services rendered). System integrity demands that agent deployers be held accountable for problems caused by misbehaving agents.

The security architecture we are developing counteracts these threats by binding each agent to a unique Agent ID (or AID). To prevent agent spoofing or masquerading, we use a certification mechanism that requires agent deployers to register their agents prior to deployment [11].

---

[1] Lookup services such as SUN's JINI are middle agents, with very restrictive matching abilities. By contrast, the LARKS language and matching algorithms we have developed allow extended semantic matching. We have identified different types of middle agents with different performance tradeoffs. Naturally, there are multiple distributed middle agents in the system.

Our security infrastructure is based upon public key cryptography. In public key cryptography, keys come in pairs--private and public. The key pair is generated using pre-defined algorithms, such as RSA and DSA. Key pairs are randomly and uniquely generated values. The private key is used to sign messages, and the public key validates that the signature of a message is a function of the private key associated with the respective public key and message. The pairing of public and private keys allows keys to be recognizable and thus operable in MAS, while ensuring that keys remain indecipherable to users.

To join a system, agents need to get a public key pair, and have it certified by an agent certification authority (CA). This public key pair and corresponding certificate are used to securely identify the agent in the community.

Within this security infrastructure, all interactions with RETSINA mediating agents (e.g., Matchmakers, Agent Name Servers) require signatures and certificates. Inter-agent interactions can also be secured through the application of security protocols, such as Netscape's SSL (Secure Socket Layer Protocol). An agent's existence at a port is verified by a challenge-response protocol to which the agent must answer. This disallows the occupation of ports by outside agents, ensuring that agents are running where they say they are.

Using encryption techniques and protocols, RETSINA has endeavored to preclude a crisis in confidence and/or security in the agent community.

4.1 Denial of Service Attacks/Socket Migration

In order to preclude such a crisis in confidence, the RETSINA Lab has also implemented mobile sockets for the TCP stack in the Unix (Linux) kernel. This is significant for a few reasons. For one, this is the first time that researchers have implemented mobile TCP sockets at the Unix kernel level. Secondly, socket migration enables network-based agents and services to strongly migrate quickly and efficiently to other computers even though they already have an established network connection with another peer. Third, socket migration technology potentially enables any service to migrate a socket, not only some services that were developed in a special, proprietary, application-level language. Four, socket migration technology is transparent to network-based services that are not designed for migration. Lastly, through this kind of socket migration technology, which permits on-the-fly renumbering of IP addresses without losing network connections, and the RETSINA discovery infrastructure, it is possible to design information and service networks that continue to function during a denial of service attack, or during other forms of malicious attacks.

5. Interoperation Infrastructure

As heterogeneous agent types and Multi-Agent Systems increase, the problem of interoperability among agents and agent systems becomes an increasingly pressing concern for developers intent upon contributing to a fully interactive and interoperable World Wide Web of Services. Interoperability is an issue for communications and cooperation between heterogeneous agents, and between heterogeneous Multi-Agent Systems. For agent-to-agent interoperability, RETSINA has developed the middle agents and corresponding agent capability language, LARKS, as well as developing DAML-S, an ontology of Web Services (see Section 3 and [10]).

As for MAS interoperability, it is unreasonable to assume that developers will abdicate their respective MASs, given that considerable investments of time, money and intellectual labor have gone into them, in order to comply with a single standard, such as FIPA. Likewise, RETSINA has operated under the premise that heterogeneous MASs will persist. As such, we have considered the options for communicating across MAS boundaries, and have developed an instance of an InterOperator, a translating service sharing the protocols and languages of two different systems, the RETSINA and the Open Agent Architecture (OAA) MAS [12]. The InterOperator is a transparent translating service, communicating in and between both systems. The InterOperator solves the problems of different ACLs, different protocols, agent registration and advertisements, as well as answering to the criterion of efficiency in extending agent system interoperability.

In connection with interoperation and the CoABS Grid, the RETSINA Lab has developed the RETSINA Jini Service-Grid Agent InterOperator Service. The notion of a Jini service is to offer a well-defined interface to some commonly used objects. The Jini Service-Grid Agent InterOperator allows CoABS Grid and Jini-based agents to access such services.

Via the RETSINA-Grid InterOperator, more than 26 RETSINA agents, in 6 RETSINA MAS applications, register with the GITI CoABS Grid LUS, and are available 24 hours a day, 7 days a week (24/7), for use by other research groups for realistic experimentation and testing. These agents can be found at: http://coabs.globalinfotek.com/coabs_private/private_pages/grid.html.

RETSINA continues to develop other instances of the InterOperator to solve the problem of MAS interoperation.

6 Relevant Applications

The RETSINA MAS Infrastructure has been applied to many knowledge and user domains, including E-commerce applications, financial portfolio manage-ment, personalized communications and display, aircraft maintenance, dynamic supply chain management, and military logistical applications, among others. Additionally, the applications of RETSINA are being extended to include navigation and communications in automobiles, which involves dynamic team and personal scheduling/planning, route planning and other information gathering and fusion, under a great variety of contingencies.

As our applications make clear, agent technology is promising to provide the kind of ubiquitous, collaborative, transparent, and non-intrusive computing that will allow software partners to contribute greatly to human endeavors in any number of activities. We include the research results of several applications in this section, and discuss few of these applications in this summary form, below.

6.1 Non-Combatant Evacuation Operation (NEO)

The NEO application is a Multi-Agent, Multi-Agent System (MAS) demonstration of agent technology in a Noncombatant Evacuation Operation (NEO) [13]. NEO illustrates the interaction of agent teams in aiding human teams to cooperatively plan and execute a hypothetical evacuation of US civilians from a Middle Eastern city in an escalating terrorism crisis. In NEO, RETSINA and Open Agent Architecture (OAA) agent teams and systems cooperate with each other and their human counterparts to evaluate a crisis situation, form an evacuation plan, follow an evolving context, monitor activity, and dynamically re-plan. NEO demonstrates the interoperability and use of two disparate agent systems' teams for aiding humans (officers and Ambassador) to effectively monitor the scenario, retrieve and fuse information for immediate use, and to plan and re-plan an emergency evacuation.

6.2 Coalition Operations: COAX

The RETSINA contribution to the CoAX TIE (for the CoAX TIE demo see http://www.aiai.ed.ac.uk/project/coax/) demonstration is the RETSINA CoAX Visualizer and the RETSINA CoAX Grid Logger. The RETSINA CoAX Visualizer permits the visualization of coalition domains, and their nested relationship to each other; agents, which are situated in "coalition domains," and may belong to multiple coalition domains; and non-agent entities, which are also situated in coalition domains, and may belong to multiple coalition domains.

The RETSINA CoAX Grid Logger provides a log facility for receiving log messages from both Grid and, if necessary, non-Grid agents; filtering log message content according to the restrictions of a coalition domain-specific security level; archiving log messages to log files according to the security level of the message; forwarding log messages to users of RETSINA CoAX Visualizers according to the security level of the message and the security level of the user. For example, if a Visualizer user operates at security level 3, for a coalition domain, then the RETSINA CoAX Grid Logger will forward messages from that coalition domain with security levels 0, 1, 2, and 3 to that user's Visualizer. A user must be granted a security level access right for each coalition domain of which he is a member.

6.3 Customer Coalitions in the Electronic Marketplace: COALA

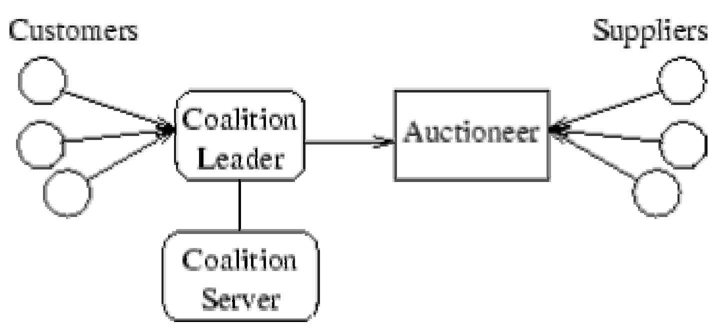

Figure 2: COALA **System Architecture**

As an initial problem domain in E-commerce, we chose collective book purchasing. In the university setting, relatively large numbers of students enrolled in the same classes purchase the same required books. Such groups can be bundled as coalitions, given the ease of payment collection, distribution of goods, and the potentially large number of users involved in the system [14]. The test-bed system (see figure 2) consists of a coalition server, an auctioneer agent, a set of supplier agents, and a web-based interface for end users. The system is based on a simple pre-negotiation protocol and a variation of sealed-bid reverse auction that allows suppliers to disclose their discount policies.

Agents use the WWW interface to conduct reverse auctions with supplier agents. The supplier agents, in turn, are given a step function volume discount schedule and make their bids according to projected sizes of coalitions. After the reverse auction is complete, the coalition server opens the coalition to new members, who can join the group if they meet the entrance requirements. After the group is formed, the coalition server proceeds to execute the transaction.

6.4 Stable and Efficient Coalitions

Auctions have become an opportunity for buyers on the Internet to form coalitions in order to negotiate with sellers and purchase various items at volume discount prices. In earlier research (see 6.3), we showed that both buyers and sellers can benefit from buyer coalitions, analyzed coalition formation models and protocols, and proposed a framework for buyer agents to form coalitions and negotiate with sellers. In this subsequent research, we went on to show how buyers could match their preferences with items and coalitions could calculate the division of surpluses (savings) in a stable manner. Our GroupBuyAuction overcomes the problems of computational complexity in dealing with thousands of buyers by forming buyers into coalitions for *categories* of items, not only for a particular item. Using *OR-asking*—a list of items indicating that a buyer would buy any one of the items under certain conditions, e.g., "I will buy camera A for $300 or less, or camera B for $400 or less"—buyers are able to request and sellers to respond to a list of items, within a particular category. Sellers present proposals for items based on volume and a leader agent within the coalition manages a reverse auction on behalf of the buyers, by virtue of their coalition buying power. The leader then splits the coalition into sub-coalitions, each of which consists of buyers preferring the same item. The leader selects the winning seller for each sub-coalition, and calculates the differential surplus division among the buyers. (Buyers may pay different prices for identical items depending on their reservation prices) [15].

6.5 Combinatorial Coalitions

Volume discounts for group buying and the need for buying complementary items exists in real markets for buyers. Economists and computer scientists have shown interest in both coalition formations and combinatorial buying, but no work had existed that dealt with both phenomena simultaneously. In this research, we consider an electronic market in which both coalition formation and combinatorial auctions exist. Each buyer is assumed to want a group of complementary goods and has a reservation cost representing the highest cost she can pay for the complementary set. The items have no utility individually and so combinatory buying is a must. Coalition formation ensures that buyers take advantage of optimal volume discounts. Such a buying problem is called a Combinatorial Coalition Formation (CCF) problem. We considered the problem of CCF and payoff division in such an electronic market and developed an algorithm that both satisfies the maximized savings for buyers and ensures stability of the payoff division amongst the members, so that each member's price is less than the price she would have paid outside of the coalition [16].

6.6 Aircraft Maintenance

Access to information is vital for mechanics doing maintenance on aircraft. Maintenance must be completed under time constraints, and a significant portion of a mechanic's time is spent looking for appropriate information from other mechanics or from paper documentation. Reports must be read and written, information sources queried and consulted, and information must be stored and organized. Not only does this take considerable time, it also results in inconsistent updates, ad hoc handwritten documentation, and lack of access to old but useful information sources. In order to address these problems, we have developed RETSINA infrastructure for use in wearable computers for mechanics' decision support during aircraft maintenance.

6.7 Mobile Communications with Heterogeneous Agents: MOCHA

In order to solve the problems arising out of the plethora of communications devices available to users and those trying to reach them, we have developed MOCHA, a multi-agent system for “anyware” communications and display--a mobile communications network that runs on any platform and reaches users anywhere. Using MOCHA, human users create personal agents who perform tasks that humans could not otherwise accomplish.

MOCHA provides a seamless, single interface for users who access a complex array of agent technology. Users are presented with a familiar, form-based interface to create personal agents. Customized user preferences form the basis of agent interactions to carry out tasks. Personal agents locate and contact other users, and help plan and schedule collaborative tasks with the personalized agents of other users.

With the benefit of personalized agents, users are provided with a single mechanism that leaves the multiple sources for contact up to the personalized agent itself. The agent decides on method of delivery based on agent knowledge of sender and receiver preferences and device availability. MOCHA's "anyware" communications and display technology also means that information can be displayed on any device in any location. Information can be displayed wherever the user goes, and on any available display platform [17].

7. Bibliography

[1] K. Sycara, M. Paolucci, M. Van Velsen, and J.A. Giampapa, The RETSINA MAS Infrastructure. In *Autonomous Agents and MAS*, Volume 7, Nos. 1 and 2, July, 2003.

[2] M. Paolucci and K. Sycara, An Exploration in MAS Scalability in *IJAS*

[3] O. Shehory, and K. Sycara K., The RETSINA Communicator. In *Proceedings of the Fourth International Conference on Autonomous Agents*, June 3-7, Barcelona, Spain: 199-200, 2000.

[4] B. Langley, M. Paolucci, and K. Sycara, Discovery of Infrastructure in Multi-Agent Systems. In *Agents 2001 Workshop on Infrastructure for Agents, MAS, and Scalable MAS.*

[5] Sycara, S. Widoff, M. Klusch and J. Lu, LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. In *Autonomous Agents and Multi-Agent Systems*, 5, 173–203, 2002.

[6] K. Sycara, Brokering and Matchmaking for Coordination of Agent Societies: A Survey. In *Coordination of Internet Agents*, A. Omicini et al. (eds.), Springer, 2001.

[7] K. Decker, K. Sycara, and M. Williamson, Middle-Agents for the Internet. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, 1997.

[8] K. Sycara, M. Klusch, S. Widoff, and J. Lu, Dynamic Service Matchmaking Among Agents in Open Information Environments. In *Journal ACM SIGMOD Record* (Special Issue on Semantic Interoperability in Global Information Systems), A. Ouksel, A. Sheth (eds.), Vol. 28, No. 1, March 1999, pp. 47-53.

[9] K. Sycara, J. Lu, M. Klusch, and S. Widoff. Matchmaking Among Heterogeneous Agents on the Internet. In *Proceedings AAAI Spring Symposium on Intelligent Agents in Cyberspace*, Stanford, USA, 1999.

[10] Sycara, K. Lu J. and Klusch M., Interoperability Among Heterogeneous Software Agents on the Internet. Technical Report CMU-RI-TR-98-22, CMU Pittsburgh, USA, October 1998.

[11] H.C. Wong and K. Sycara, Adding Security to Multi-Agent System Adding Security and Trust to Multi-Agent Systems. In *Proceedings of Autonomous Agents '99 Workshop on Deception, Fraud, and Trust in Agent Societies*, May, 1999, pp. 149 - 161.

[12] J. Giampapa, M. Paolucci, and K. Sycara, Agent Interoperation Across Multiagent System Boundaries. In *Proceedings of Agents 2000*, Barcelona, Spain, June 3-7, 2000.

[13] J. Giampapa, M. Paolucci, and K. Sycara, Agent Interoperation Across Multiagent System Boundaries. In *Proceedings of Agents 2000*, Barcelona, Spain, June 3-7, 2000.

[14] M. Tsvetovat, K. Sycara, Y. Chen, and J. Ying, Customer Coalitions in the Electronic Marketplace. In *Agents 2000 Conference*, Barcelona, Spain, June 3- June 8, 2000.

[15] J. Yamamoto and K Sycara, A Stable and Efficient Buyer Coalition Formation Scheme for E-Marketplaces. Presented at the Fifth International Conference on Autonomous Agents (Agents 2001) in Montreal Canada, May 28-June 1, 2001.

[16] C. Li and K. Sycara, Algorithm for Combinatorial Coalition Formation and Payoff Division in an Electronic Marketplace. Presented at the *First International Joint Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS 2002), July 2002, Bologna, Italy.

[17] O. Shehory, K. Sycara, G. Sukthankar, and V. Mukherjee, Agent Aided Aircraft Maintenance Agent Aided Aircraft Maintenance. Presented at the Third International Conference on Autonomous Agents, May, 1999.